A step toward reusable model fragments

H. Ahriz[1], and M. Tomasena[2]
[1] School of Computer and Mathematical Sciences, Robert Gordon University,
St Andrew Street, Aberdeen AB25 1HG, U.K.
E-mail: ha@scms.rgu.ac.uk
[2] SysCom, Université de Savoie,
Le Bourget du Lac, 73376, France.
E-mail: tomasena@univ-savoie.fr

Abstract

In this paper we describe a system to elaborate models which are suitable for model based reasoning. A set of model fragments selected from a library will be put together to build a model candidate. The system relies on the bond graph notation, which allows a uniform approach for the different physical domains and offers a compositional view of the system. Modeling requires the exploration of a search space of potential model candidates. These models are checked to be consistent with a set of behavior constraints and modeling hypotheses provided by the user.

1. Introduction

Most of the model-based reasoning systems do not pay much attention to model construction, and it was generally accepted that a model was available or could be easily obtained. This assumption is not always realistic. Moreover, the importance of using a good model is obvious because building a model is the starting point in the whole process [1, 2, 3, 4 and 5]. Traditionally, models are constructed by hand and are then used in experiments to ensure acceptable results. Models produced in this manner tend to include everything, including issues irrelevant to an application, and require solid competencies in mechanical, hydraulics, electricity and thermodynamics. These considerations indicate the need for new approaches to modeling, based on more rigorously defined modeling processes.

Few research works have already addressed this issue. GoM (Graph of Models) of Addanki [6], which represents a collection of models built by an expert in a particular domain. The collection is represented in terms of a graph in which each node represents a model, whereas an edge is labeled with an assumption (simplification or refinement). The Prompt system of Weld [1] allows for navigation through this kind of graphs. A most significant work is CM (Compositional Modeling) of Falkenhainer and Forbus [3], which is devoted to generate qualitative and quantitative answers to queries about physical systems. Other works have been derived from the latter, namely "Automated Model Selection for Simulation" of Iwasaki and Levy [7] and "Causal Approximations" of Nayak [4]. The works of Amsterdam [8] and Biswas and Yu [9] are also related to ours since they use bond graphs as a modeling language. The main distinction of our approach is its non-deterministic nature. Actually we consider that modeling process requires the exploration of a search space. This search space could have several solutions (i.e. models), could accept several cost criteria and could be explored with various search strategies. The explicit use of modeling hypotheses and behavior constraints is a means to limit the exploration of the search space and select and adequate candidate.

Our work intends to introduce more automation in the different modeling tasks, and led to the system: AIMD (Automated Intelligent Modeler for Diagnosis). Modeling and diagnosis are the two main functions of AIMD. In this article we focus on modeling, the diagnosis process is outside the scope of the present article. Causal-model based diagnosis is treated by several researches like: Ahriz and Xia, [10]; Console et al. [11] and Mosterman and Biswas [12].

2. Modeling elements

In this paper we consider the following pump system as a case study (figure 1): a motor is driven by a voltage source and, in turn, drives a pump, which pumps a fluid from tank 1 to tank 2.

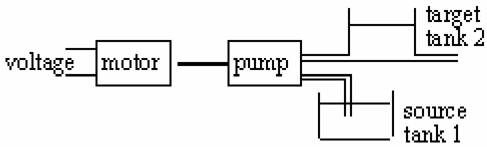


Figure 1 — Case study system

The modeling process is based upon the consideration of two groups of inputs: the scenario dependent ones, and the scenario independent ones (by scenario we mean the modeling session tackled by the designer). Scenario dependent inputs are the description of the physical system, a set of modeling hypotheses and a set of behavior constraints, whereas scenario independent ones are a library of generic model fragments, along with other generic knowledge concerning physical systems. Given such entries, AIMD is able to elaborate a parsimonious model representing the system. This model is described in terms of a bond graph, a set of qualitative and quantitative equations and a causal graph.

The modeling process relies on the Bond Graphs. This formalism, introduced by Rosenberg and Karnopp [13], is based on modeling the energy flow/power between the system components and inherently enforces continuity of power and conservation of energy. This provides a systematic framework for building consistent and well-constrained models of physical systems across multiple domains (e.g., electrical, mechanical, hydraulic). Bond graphs allow for compositional modeling and make models applicable to qualitative processing. This renders them useful in situations where precise numerical information may not be available. However, analytic system models derived from bond graphs are also amenable to quantitative simulation. Furthermore, bond graphs embody a direct relation between state variables and physical component parameters, and their causality constraints provide the mechanisms for effective diagnosis.

As we pointed out, the modeling approach aims to be modular and declarative. On the other hand the nature of modeling is intrinsically non-deterministic. For these reasons, AIMD was implemented in Prolog. This choice allows for a declarative representation of the different kinds of knowledge in terms of logical relations and is naturally adapted for the exploration of a search space. Moreover, the performance of AIMD in terms of execution time showed to be encouraging, although this is not an issue since the model construction is made off line.

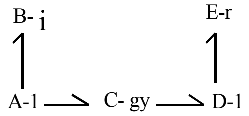
3. Scenario-independent inputs

Ideally, a library of generic components should consist of "context-free" component models that adhere to the "no function in structure" principle stated by de Kleer and Brown [14]. The definition of the library of components respects this principle. This is possible since the modeling process takes into account, explicitly, other sources of knowledge. That means that for a given component the fragment selection task could pick up one specific fragment of model in the library even if this selection is aberrant from a global point of view. That doesn't matter, since the fragment will be ruled out in the successive steps and other fragments at higher level of complexity (detail) of the same component will be considered. Each component, in a given domain, has one or more associated model fragments, from the simplest one to a most complex one. Complexity is defined as the number of bond graph elements from which a model fragment is made. It forms a partial order relation.

For example, a motor can be represented by 5 model fragments, one of complexity 1 (GY), one of complexity 2 (GY+R), two of complexity 3 (GY+R+C and GY+R+I) and finally one of complexity 4 (GY+R+C+I). The previous elements stand for GY= gyrator, R= coil resistance, C= coil capacitance, I= coil inductance.
The complexity of a whole model, will be the sum of the complexities of all its fragments.
Each fragment is represented by:

- a name: the same one must be used for the component in the device description;
- a domain: simple physical domain, or joined domains to represent a transformation from one domain to another as in the example of the motor;
- an integer representing the fragment complexity;
- a description (i.e. bond graph) consisting of: (a) input and output of the bond graph, in order to be linked to other fragments, (b) the list of Prolog variables each of which refers to a node in the bond graph, and, finally, (c) a list of bonds between nodes. A node is represented by a couple name-type, where name is a Prolog variable and type is the type of the element (i, c, r, gy, tf, se, sf) or the junction (1, 0). For example the fourth fragment of the motor corresponds to the bond graph shown in figure 2.

As we mentioned before, a component may have several model fragments corresponding to various situations of use, and depending on the presence or not of particular physical phenomena.



component(motor, electric-mechanical, 3,
description(input(A),output(D), [A,B,C,D,E],
[bond(A-1,B-i), bond(A-1,C-gy),
bond(C-gy,D-1), bond(D-1,E-r)])).

Figure 2 — Fourth fragment model of the motor

In each model fragment there is an indication of the modeling hypothesis implicitly used. This indication is obtained from the elements present in the bond graph. All we need to do, thus, is to provide a correspondence between these elements and the physical phenomena. This is achieved by defining a set of relations «corresponds» like:

```
corresponds(friction,          r).
corresponds(dissipation,       r).
corresponds(compressibility,   c).
```

Furthermore, we don't need the library to contain, for a particular component, all the possible model fragments. A tube, for example, can be represented either by one of the following elements: C, I, or R, which correspond to three modeling hypotheses. These hypotheses do mean respectively: H1: compressible fluid (flexible walls); H2: inviscid liquid (long and narrow tube); H3: viscous liquid (rough walls). It is unnecessary to encumber the library with other fragments, which can be obtained by combining the basic ones. This choice constitutes an improvement compared to the graph of models in Addanki [6], which enumerates and represents all the combinations.

4. Scenario dependent inputs

The device structure. The device structure representation is an abstracted view (figure 3) of the physical system. It is Component-Connection oriented, and, thus, contains the description of the system components, relations (including connections) between component terminals, and the specification of the inputs as well as the outputs of the system.

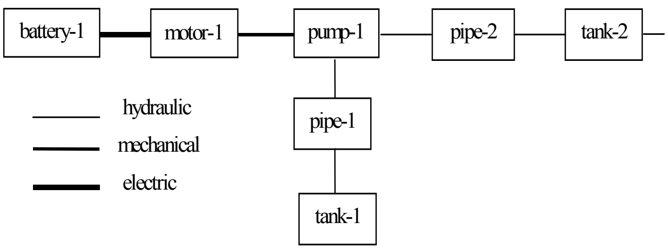


Figure 3 — Schematic description of the case study system

The declarative description of device structure is:

```
input ([]).
output (tank2-hydraulic).
set_of_relations([
connection(electric,   [battery-1],[motor-1]),
connection(mechanical, [motor-1],  [pump-1]),
connection(hydraulic,  [pump-1],   [pipe-2]),
connection(hydraulic,  [pipe-1],   [pump-1]),
connection(hydraulic,  [tank-1],   [pipe-1]),
connection(hydraulic,  [pipe-2],   [tank-2]) ]).
```

In addition to the “connection” relation, other kinds of relations can be used in AIMD, to allow someone to represent relations such as a heat transfer.

*Modeling Hypotheses***.** The variety of model fragments of each component are due to the various modeling hypotheses one can consider when representing a physical system. The user is allowed to state explicitly such modeling hypotheses about the device at hand: an a-priori set can be stated, using “consider” relations like in the work of Falkenhainer and Forbus [3]. For example: “consider the friction in the motor”, is represented by:

```
consider(mechanical, friction, motor-1).
```

In our case study, if the user wants to take account of the friction in the motor, then AIMD will consider only the model fragments associated to the motor component in the mechanical domain that contain the element "r" in their description.

Behavior Constraints. In addition to the description of the system’s structure and the modeling hypotheses, inputs could include a set of behavior constraints. A behavior constraint describes, in qualitative terms, one possible dynamic behavior of some device variables. The representation of these expected behaviors is done through a “constraint” predicate:

```
constraint(&lt;component&gt;, &lt;variable&gt;, &lt;segment&gt;).
```

It specifies the physical component and the concerned variable within it, as well as an ordered list of couples (value, derivative) for this variable, describing its expected dynamic behavior in qualitative terms (segment). The qualitative space considered is {-, 0, +). For example, the following constraint: “when the source tank becomes empty, the motor speed increases”, is represented by the couple of constraints:

```
constraint(motor-1, speed, [(0,+),(+,+),(+,+)]).
constraint(tank-1, volume, [(0,−),(−,−),(−,0)]).
```

Each segment is a succession of time points and time intervals, such as in QSIM (Kuipers [15]). All the couples of values for each variable are represented for the same times. Each time point has the value 0 for at least one variable or its derivative. This point time corresponds to a sign change or the evolution of a variable.

5. Modeling Process

The modeling process consists on the three following tasks:

*1. Selection of model fragments***.** The inputs to this task are the structural description of a system to be modeled and a set of modeling hypotheses. For each component in a given domain, the model selection procedure consists in choosing the simplest model that doesn't contradict the set of the modeling hypotheses. Initially, this set may include an a-priori list of explicit modeling hypotheses; otherwise, the selection procedure takes the simplest model of each component. Successive selections are made increasing the degree of complexity of fragments starting with the least complex ones. If we consider a device with n components, and an average number of model fragments of p, then the search space will cover all the p^n combinations. Fortunately, these combinations are not explored totally, and AIMD allows for the application of two search strategies. The modeling process could produce either:

- The first parsimonious model (least complex) satisfying the criteria, by the application of a branch and bound search, or
- The first model (not necessarily parsimonious) satisfying the criteria by the application of a depth-first search. The depth-first search means that one component, at a time, has to be made more complex.

*2. Fragment Assembling***.** Fragment assembling is made according to the structural description and some compositional rules proper to the bond graph formalism (due to limitation of space we cannot elaborate on these rules). Once the assembling is performed, AIMD tries to assign the causality bars to the bond graph representing the whole model. This procedure, described by Rosenberg and Karnopp [13] and by Top and Akermans [16], may lead to two cases:

- a conflict of causalities: we must, thus, loop (backtrack) to the selection task to pick up other fragments (we choose a more complex fragment for one component);
- the procedure is successful: we continue with the next task (if there are many solutions, we have to cope with all these possible models.

3. Model verification. In a nutshell, the purpose of verification is to get confident about the device model. This is crucial when handling the diagnosis task: when a discrepancy between what is observed and what is intended is detected, there is no doubt that some-thing is wrong with the device, so we never incriminate the model in use.

For the purpose of verification, a set of qualitative differential equations is derived from the bond graph. We can now provide the following definition: A model is said to satisfy behavior constraints, if we find a matching between one of the possible simulated behaviors and the expected one. In order to be able to compare simulated behaviors with the expected one, AIMD uses a table of generic correspondences between external variables used to state the behavior constraint (like speed) and the internal variables of the bond graph (like "f"). Some or this correspondences are described in table1.

We use a QSIM [15] like simulation in order to simulate the behaviors of a model. Adapting to the bond graph formalism, we elaborated the following qualitative differential equations (QDEs):

- add(Y, [(X_1, s_1), …, (X_n, s_n)]): represents the sum of efforts or flows in a junction, $Y = \sum (s_i) X_i$, where s_i are the signs (+, –) of each X_i variable;
- equal([X_1, …, X_n]): represents the equality of efforts or flows in a junction, $X_1 = \ldots = X_n$;
- int(Y, X): integration relation used for a C or I element (Y and X are either efforts or flows), $Y = \int X\, dt$;
- mon(Y, X): a monotonic function used for a R, TF or GY element.

We adopt the alternation between time points and time intervals [15], and adapt it to the qualitative variables domain {–, 0, +}. As a result, we obtain 15 P-transitions and 15 I-transitions which are valid between each ([x], [x'])$_1$ state and a next ([x], [x'])$_2$ state.

```
correspondence(hydraulic,  pressure,    e).
correspondence(hydraulic,  flow,        f).
correspondence(mechanical, force,       e).
correspondence(mechanical, speed,       f).
correspondence(electric,   tension,     e).
correspondence(electric,   courant,     f).
correspondence(thermal,    temperature, e).
correspondence(thermal,    energy_flow, f).
```

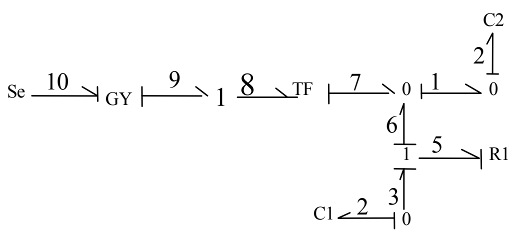
TABLE 1— Correspondence between bond graph elements and physical entities

6. Results

For our case study, let us consider an execution with the following behavior constraints:

```
constraint(pressure, tank-1,     [(0, -),(-, -),(-, 0)]).
constraint(flow,     tank-2,     [(0, +),(+, +),(+, 0)]).
```

Figure 4 represents the produced model in terms of a bond graph and in terms of a set of qualitative equations.



int(e-2, f-2)	add(f-1,[(f-6,+),(f-7,+)])
int(e-4, f-4)	equal([e-1, e-6, e-7])
mon(f-5, e-5)	add(e-9, [(e-8, +)])
add(f-2, [(f-1, +)])	equal([f-8, f-9])
equal([e-1, e-2])	mon(f-7, f-8)
add(f-4, [(f-3, -)])	mon(e-8, e-7)
equal([e-3, e-4])	mon(f-9, e-10)
add(e-5, [(e-3, +),(e-6, -)])	mon(f-10, e-9)
equal([f-3, f-5, f-6])	

Figure 4 — Case study model

Note that C1 and C2 represent tank 1and tank 2, R1 and R2 represent pipe1 and pipe 2, TF represents the pump and GY the motor, whereas Se is used for the voltage. Each bond has a number *n* which is to be associated with internal variables *e-n* and *f-n*.

Conclusion

The compositional point of view of the modeling task is the basis of our modeling framework. This approach requires, first, to break a physical system into smaller parts (components) and then to assemble the system model from the model fragments of the parts. Bond graph modeling greatly facilitates this requirement since it offers a uniform formalism for the definition of generic component models. We believe that this represents an important step towards a library of reusable models. The nature of our modeling approach is intrinsically non-deterministic and requires the exploration of a search space. Different models are checked to be consistent with a set of behavior constraints and modeling hypotheses provided by the user. This task of model verification, and particularly the qualitative simulation, ensures that the model at hand is reliable for a fault diagnosis task.

References

[1] D.S. Weld, Reasoning about Model Accuracy, Artificial Intelligence, 56, pp. 225-300, 1992.

[2] P. Dague, O. Raiman and P. Deves, Troubleshooting: when modeling is the difficulty. Proceedings of the 6th AAAI, pp. 600-605, 1987.

[3] B. Falkenhainer and K.D Forbus, Compositional Modeling: finding the right model for the job. Artificial Intelligence 51, p. 95-143, 1991.

[4] P.P Nayak, Causal Approximations. Artificial Intelligence 70, p. 277-334, 1994.

[5] S. Xia, D.A. Linkens and S. Bennett, Automatic Modelling and Analysis of Dynamic Physical Sys-tems using Qualitative Reasoning and Bond Graphs. Intelligent Systems Engineering Journal, pp. 201-212, 1993.

[6] S. Addanki, R. Cremonini, J.S. Penberthy, Graphs of Models. Artificial Intelligence, 51, pp. 145-177, 1991.

[7] Y. Iwasaki and A.Y. Levy, Automated Model Selection for Simulation. KSL-93-11 Report, Stanford University, 1993.

[8] J. Amsterdam, Automated Qualitative Modelling of Dynamic Physical Systems. PhD Thesis, MIT AI Lab.

[9] G. Biswas and X.Yu, A Formal Modeling Scheme for Continuous Systems: Focus on Diagno-sis. Proceedings of IJCAI-93, pp.1474-1479, 1993.

[10] H. Ahriz and S. Xia, Automatic Modelling for Diagnosis. QR97, Cortona, Italy, p. 3-13, June, 1997.

[11] L. Console, D.Dupre and P. Torasso, A theory of diagnosis for incomplete causal models. Pro-ceedings of IJCAI-89, pp. 1311-1317, 1989.

[12] P. J. Mosterman and G. Biswas, An integrated architecture for model-based diagnosis. 7th Principles of Diagnosis workshop, pp.167-174, Val Morin, Canada, 1996.

[13] R. Rosenberg and D. Karnopp, Introduction to Physical System Dynamics. McGraw-Hill, NY, 1983.

[14] J. de Kleer and J.S. Brown, A Qualitative Physics Based on Confluences. Artificial Intelligence 24, pp.7-83, 1984.

[15] B.J. Kuipers, Qualitative Simulation. Artificial Intelligence, 29, 3, pp.289-338, 1986.

[16] J. Top and H. Akermans, Computational and Physical Causality . Proceedings of IJCAI-91, pp.1171-1176, 1991.